



# UNITED STATES PATENT AND TRADEMARK OFFICE

TD

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 10/091,237      | 03/04/2002  | Hong Su              | 10013661-1          | 7188             |

7590 10/16/2006

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort Collins, CO 80527-2400

EXAMINER

STEVENS, ROBERT

ART UNIT PAPER NUMBER

2162

DATE MAILED: 10/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

10/091,237

Applicant(s)

SU ET AL.

Examiner

Robert Stevens

Art Unit

2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 31 July 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-26 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-26 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. The Office withdraws the objection to the specification, as omitting a Federally Sponsored Research Statement, in light of the amendment. The Office maintains the previous rejections of the claims under 35 USC §103(a), in light of the amendment.

### *Response to Arguments*

2. Applicant's arguments have been fully considered but they are not persuasive.

### Note By Applicant

Applicant asserts on page 9 of the Amendment in the section entitled "Office Notes" that the filing of an RCE on 12/15/2005 was prior to the reception of the non-final action mailed 11/28/2005.

The Office maintains that the status of the case was appropriate for the actions taken by the Office.

### Specification Objection

Applicant asserts on pages 9-10 of the Amendment in the section entitled "Specification" that that a Federally Sponsored Research and Development statement, as set forth in MPEP §310, is not necessary because "there was no federally sponsored research and development".

This objection, found on page 3 of the Final Action, mailed 4/26/2006 stated:

7. The specification is objected to because a Federally Sponsored Research and Development statement, as set forth in MPEP §310, appears to be necessary. On page 29 of the Power Point briefing [Su, Hong, et al., "Automating Transformation of XML Documents", WIDM 2001

Powerpoint Presentation, Atlanta, Ga, Nov. 2001, pp. 1-29] presented at the WIDM 2001 Conference by the inventors, mention is given to the involvement of the NSF (i.e., a US Government agency that provides R&D funding to researchers). Note that this briefing (a copy of which was provided with the Office action mailed 11/28/2005 and cited in the 11/28/2005 Form PTO-892) discloses the subject matter of the WIDM 2001 paper that is the subject of the inventors' Rule 131 affidavits concerning reduction to practice.

The Office withdraws the previous objection to the specification, in light of Applicant's statement.

Rejections under 35 USC §103(a)

Regarding claims 1 -17, including independent claims 1 and 10, Applicant further asserts on pages 12-13 that the rejections set forth under 35 USC §103(a) are improper because the cited references, ShemaMatching and XEM, are "prior art under 102(e)", and that "the disclosure relied upon is Applicant's own work".

The Office respectfully disagrees. Applicant's arguments hinge upon the cited references being 102(e) art. It is noted however, that "102(e) art" consists of US patents and US patent applications (and certain international patent applications). The cited references, ShemaMatching and XEM, are research papers. These references are not "102(e) art", and thus Applicant's arguments fail. Additionally, Applicant supplied no evidence to support the assertion that the cited references are Applicant's own work.

Art Unit: 2162

Regarding claims 2 and 17-21, Applicant asserts on page 13 of the Amendment that the claims are not rendered obvious over SchemaMatching in view of XEM and further in view of Swamy.

The Office respectfully disagrees. The Office re-asserts the rejections of these claims as set forth below, and in the previous action, mailed 4/26/2006, noting that Applicant has merely stated that a disagreement exists but provided no arguments in support of Applicant's position.

Regarding claim independent 18, Applicant asserts on page 14, that the rejections set forth under 35 USC §103(a) are improper because the cited references, ShemaMatching and XEM, are "prior art under 102(e)", and that "the disclosure relied upon is Applicant's own work". Applicant further asserts on pages 13-14 of the Amendment that claim 18 is not rendered obvious over SchemaMatching in view of XEM and further in view of Swamy.

The Office respectfully disagrees. Applicant's arguments hinge upon the cited references being 102(e) art. It is noted however, that "102(e) art" consists of US patents and US patent applications (and certain international patent applications). The cited references, ShemaMatching and XEM, are research papers. These references are not "102(e) art", and thus Applicant's arguments fail. Additionally, Applicant supplied no evidence to support the assertion that the cited references are Applicant's own work. The Office further re-asserts the rejection of this claim as set forth below, and in the previous action, mailed 4/26/2006, regarding the incorporation of the Swamy reference, noting that Applicant has merely stated that a disagreement exists but provided no arguments in support of Applicant's position.

Regarding claims 2, 17 and 19-26, Applicant further asserts on page 15 of the Amendment that these dependent claims are patentable over the cited prior art for the reasons asserted above.

The Office respectfully disagrees. See the Office's counterarguments above.

Regarding claims 5-9, Applicant asserts on pages 15-16, that the rejections set forth under 35 USC §103(a) are improper because the cited references, ShemaMatching and XEM, are "prior art under 102(e)", and that "the disclosure relied upon is Applicant's own work". Applicant further asserts on page 13-14 of the Amendment that claims 2 and 17-21 are not rendered obvious over SchemaMatching in view of XEM and further in view of Buneman.

The Office respectfully disagrees. Applicant's arguments hinge upon the cited references being 102(e) art. It is noted however, that "102(e) art" consists of US patents and US patent applications (and certain international patent applications). The cited references, ShemaMatching and XEM, are research papers. These references are not "102(e) art", and thus Applicant's arguments fail. Additionally, Applicant supplied no evidence to support the assertion that the cited references are Applicant's own work. The Office further re-asserts the rejection of this claim as set forth below, and in the previous action, mailed 4/26/2006, regarding the incorporation of the Buneman reference, noting that Applicant has merely stated that a disagreement exists but provided no arguments in support of Applicant's position.

Art Unit: 2162

Regarding claims 22-26, Applicant asserts on pages 16-17, that the rejections set forth under 35 USC §103(a) are improper because the cited references, SchemaMatching and XEM, are “prior art under 102(e)”, and that “the disclosure relied upon is Applicant’s own work”.

Applicant further asserts on page 13-14 of the Amendment that claims 2 and 17-21 are not rendered obvious over SchemaMatching in view of XEM and further in view of Swamy and Buneman.

The Office respectfully disagrees. Applicant’s arguments hinge upon the cited references being 102(e) art. It is noted however, that “102(e) art” consists of US patents and US patent applications (and certain international patent applications). The cited references, SchemaMatching and XEM, are research papers. These references are not “102(e) art”, and thus Applicant’s arguments fail. Additionally, Applicant supplied no evidence to support the assertion that the cited references are Applicant’s own work. The Office further re-asserts the rejections of these claims as set forth below, and in the previous action, mailed 4/26/2006, regarding the incorporation of the Swamy and Buneman references, noting that Applicant has merely stated that a disagreement exists but provided no arguments in support of Applicant’s position.

For these reasons, the Office asserts the rejections of the claims as set forth below.

***Continued Examination Under 37 CFR 1.114***

3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 1/16/2006 has been entered.

***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. **Claims 1, 3-4 and 10-16 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM").



**Independent claim 1 states:**

*A method of document transformation comprising:*

- a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;*
- b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and*
- c) generating a sequence of transformation operations that transforms said source tree to said target tree.*

Regarding these limitations ...

- a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;*
- b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and*

SchemaMatching discloses DTD schema matching in the Abstract, discussing the matching of elements between two DTD schemas. SchemaMatching further discusses in Example 1 of page 2, the modeling of DTD schemas for a customer (i.e., source) and a client (i.e., target) as DTD graphs. Figure 1 of page 5 shows (arranged as tree data structures) the element graphs for the customer and client DTDs, and section “2.3 Construction of a Directed Acyclic Graph (DAG)” describes the process of creating the DAG, or tree, data structure.

- c) generating a sequence of transformation operations that transforms said source tree to said target tree.*

However, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section “4 Completeness of DTD Change Operations” on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., “F()”). XEM further discloses a working framework, dubbed

Art Unit: 2162

“Marrow”, the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section “5 System Implementation: MARROW” and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section “3.2 DTD Change Primitives”.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Regarding dependent claim 3**, SchemaMatching discloses matching leaf vertices (i.e., nodes of a tree) in section “3.1 Initial Leaf Vertices Matching” on pages 5-6, discussing operational steps for “Matching Criterion 1” between leaf nodes. Discussion of “Matching Criterion 2” in section “4 Detection of Hierarchically Equivalent Elements” on page 7, discloses a stricter set of rules for matching leaf nodes that includes consideration of the tree hierarchy to the nodes being matched.

**Regarding dependent claim 4**, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section “4 Completeness of DTD Change Operations” on

Art Unit: 2162

pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Independent claim 10 states:**

*A method of document transformation comprising:*

- a) modeling a source schema of XML and a target schema of XML as a tree structure creating source tree and a target tree, said source tree having a plurality of source nodes, said target tree having a plurality of target nodes; and*
- b) generating a sequence transformation operations that transforms said source XML document to said target XML document, wherein said plurality of source nodes of said source schema are matched and transformed to said plurality of target nodes in said target schema.*

Regarding these limitations ...

***a) modeling a source schema of XML and a target schema of XML as a tree structure creating source tree and a target tree, said source tree having a plurality of source nodes, said target tree having a plurality of target nodes; and***

***wherein said plurality of source nodes of said source schema are matched and transformed to said plurality of target nodes in said target schema.***

SchemaMatching discloses DTD schema matching in the Abstract, discussing the matching of elements between two DTD schemas. SchemaMatching further discusses in Example 1 of page 2, the modeling of DTD schemas for a customer (i.e., source) and a client (i.e., target) as DTD graphs. Figure 1 of page 5 shows (arranged as tree data structures) the element graphs for the customer and client DTDs, and section “2.3 Construction of a Directed Acyclic Graph (DAG)” describes the process of creating the DAG, or tree, data structure. SchemaMatching discloses matching leaf vertices (i.e., nodes of a tree) in section “3.1 Initial Leaf Vertices Matching” on pages 5-6, discussing operational steps for “Matching Criterion 1” between leaf nodes. Discussion of “Matching Criterion 2” in section “4 Detection of Hierarchically Equivalent Elements” on page 7, discloses a stricter set of rules for matching leaf nodes that includes consideration of the tree hierarchy to the nodes being matched. SchemaMatching also discloses three exemplary methods of transforming DTD elements on page 3.

***b) generating a sequence transformation operations that transforms said source XML document to said target XML document,***

However, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section “4 Completeness of DTD Change Operations” on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., “F()”). XEM further discloses a working framework, dubbed

Art Unit: 2162

“Marrow”, the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section “5 System Implementation: MARROW” and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section “3.2 DTD Change Primitives”.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Regarding dependent claim 11**, SchemaMatching discloses matching nodes and selecting a best matching plan in phases numbered 3 and 4 on page 2 (immediately above the section labeled “2 DTD Data Model”). SchemaMatching also discloses distance (i.e., cost) and element overlap (i.e., data loss) in Example 4 of page 7, and the paragraph preceding Example 4, which discuss computing a number reflective of the amount of overlap of two graphs.

**Regarding dependent claim 12**, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section “4 Completeness of DTD Change Operations” on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD

(e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", the implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Regarding dependent claim 13**, SchemaMatching discloses the use of source and target DTDs in phase number "1." above the section entitled "2 DTD Data Model", discussing modeling source and target DTDs as graphs.

**Regarding dependent claim 14**, SchemaMatching discloses simplifying DTDs into a reduced DTD in section “2.1 Simplification Transformation on DTD”, discussing, for example, a transformation that folds a group into a flattened representation.

**Regarding dependent claim 15**, SchemaMatching discloses simplifying DTDs into a reduced DTD in section “2.1 Simplification Transformation on DTD”, discussing, for example, a transformation that merges sub-elements having the same name in a content model.

**Regarding dependent claim 16**, SchemaMatching does not explicitly disclose constraining node operations. XEM, though, teaches the well-known use of quantifier node notation, such as qmark “?”, asterisk “\*” and plus “+” in sub-section 2. Constraint Node” on page 3. XEM further discusses labelling in the third paragraph below the section header “2.2 The DTD Data Model”, discussing the labelling function *l* (i.e., “ell”). XEM discloses on pages 3-6 various operations performed on DTD models. In section “2.2 The DTD Data Model”, XEM sets forth how nodes are labelled and the notation used by one skilled in the art at the time of the invention. Section “3.2 DTD Change Primitives” sets forth various operations using that notation, including folding or flattening (see page 5 “5. flattenGroup(E, pos)”) [it being obvious that one performed the inverse of folding in order to unfold], and the changing of attributes [including relabeling] in section “3.2.3 Changes to an Attribute Type Definition” on page 6.

When certain functions are performed is merely an obvious variant.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have

Art Unit: 2162

allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

6. **Claims 2 and 17-21 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM") and further in view of Swamy et al. (US Patent No. 6,874,141, filed Jun. 29, 2000 and issued Mar. 29, 2005, hereafter referred to as "Swamy").

**Regarding dependent claim 2**, SchemaMatching does not explicitly disclose the generation of XSLT stylesheets. Swamy, though, teaches generation of XSLT stylesheets in the Abstract, discussing the generation of XSL code representation (and an XSLT stylesheet) of a mapping between a source and a target schema.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Swamy for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to compile a graphical representation of data transformations into



an XSL stylesheet representation of the mapping, as taught by Swamy in col. 3 lines 19-25.

These references were all applicable to the same field of endeavor, i.e., XML programming.

**Claim 17** is substantially similar to claim 2, and therefore likewise rejected.

**Independent claim 18 states:**

*A computer system comprising:*

*a processor; and*

*a computer readable memory coupled to said processor and containing program instructions that, when executed, implement a method of document transformation comprising:*

*a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;*

*b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and*

*c) generating a sequence of transformation operations that transforms said source tree said target tree.*

Regarding these limitations ...

*a) modeling source XML document corresponding to a source schema as a source tree having a plurality of source nodes;*

*b) modeling target XML document corresponding to a target schema as a target tree having a plurality of target nodes; and*

SchemaMatching discloses DTD schema matching in the Abstract, discussing the matching of elements between two DTD schemas. SchemaMatching further discusses in Example 1 of page 2, the modeling of DTD schemas for a customer (i.e., source) and a client (i.e., target) as DTD graphs. Figure 1 of page 5 shows (arranged as tree data structures) the

Art Unit: 2162

element graphs for the customer and client DTDs, and section “2.3 Construction of a Directed Acyclic Graph (DAG)” describes the process of creating the DAG, or tree, data structure.

*c) generating a sequence of transformation operations that transforms said source tree to said target tree.*

However, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section “4 Completeness of DTD Change Operations” on pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., “F()”). XEM further discloses a working framework, dubbed “Marrow”, that implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section “5 System Implementation: MARROW” and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section “3.2 DTD Change Primitives”.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

*a processor; and  
a computer readable memory coupled to said processor and containing program instructions that, when executed, implement a method of document transformation comprising;*

SchemaMatching does not explicitly disclose a hardware implementation environment. Swamy, though, teaches a hardware implementation environment in Figure 15, disclosing a processor (#521) and memory units (#522, 527, 529, 531) coupled via a bus (#523) for executing applications (#536), including source to target schema transformations (Abstract).

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Swamy for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to compile a graphical representation of data transformations into an XSL stylesheet representation of the mapping, as taught by Swamy in col. 3 lines 19-25. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Claim 19** is substantially similar to claim 2, and therefore likewise rejected.

**Regarding dependent claim 20**, SchemaMatching discloses matching leaf vertices (i.e., nodes of a tree) in section “3.1 Initial Leaf Vertices Matching” on pages 5-6, discussing operational steps for “Matching Criterion 1” between leaf nodes. Discussion of “Matching Criterion 2” in section “4 Detection of Hierarchically Equivalent Elements” on page 7, discloses a stricter set of rules for matching leaf nodes that includes consideration of the tree hierarchy to the nodes being matched.

**Regarding dependent claim 21**, SchemaMatching does not explicitly disclose the generation of a sequence of transformation operations. XEM, though, teaches the application of a series of transformation operations in section “4 Completeness of DTD Change Operations” on

Art Unit: 2162

pages 6-7, discussing a proof for the generation of a target DTD (e.g., G') from a source DTD (e.g., G) via a finite sequence of operations (e.g., "F()"). XEM further discloses a working framework, dubbed "Marrow", that implemented the concepts disclosed by XEM, and was demonstrated at the ACM SIGMOD 2000. (See page 7 section "5 System Implementation: MARROW" and Footnote 1.) Additionally, XEM discusses the application of DTD change primitives to child nodes in order to ensure the validity of the target DTD in section "3.2 DTD Change Primitives".

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching in view of Swamy, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

7. **Claims 5-9 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM") and further in view of Peter Buneman et al ("UnQL: A Query

Art Unit: 2162

Language and Algebra for SemiStructured Data Based on Structural Recursion”, The VLDB Journal, Issue No. 9, Springer-Verlag, © 2000, pp. 76-110, hereafter referred to as “Buneman”).

**Regarding dependent claims 5-6 and 8-9**, SchemaMatching discloses matching source and target nodes and generating a transformation between nodes in phases 3 and 4 on page 2, discussing the matching likelihood of component pairs” (phase 3) and producing a “best matching plan” in phase 4. However, SchemaMatching does not explicitly disclose computing for a sequence of transformation operations, such as unfolding. Buneman, though, teaches the calculation of value equivalence in performing operations on trees in section “4.1 value Equivalence” starting on page 88, discussing the determination of a value equivalence for a data graph d and the resulting graph from an unfold operation in the first and second paragraphs under the section heading.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Buneman for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to implement a value-based semistructured data model, as taught by Buneman in last paragraph on page 77. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Regarding dependent claim 7**, SchemaMatching does not explicitly disclose matching node labels. XEM, though, teaches the use of node labels in section “2.2 The DTD Data Model”, discussing a labeling for node attributes. It was implicit in node matching that at least one node

Art Unit: 2162

attribute must be matched, and it was a obvious variant at the time of the invention as to which attribute (e.g. label name) was matched between nodes.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching in view of Buneman, because to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.

8. **Claims 22-26 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Hong Su et al. ("Identification of Syntactically Similar DTD Elements for Schema Matching", The Second International Conference on Web-Age Information Management (Waim 2001), Xi'an, China, July 2001, pp. 1-13, hereafter referred to as "SchemaMatching") in view of Hong Su et al. ("XEM: Managing the Evolution of XML Documents", Eleventh International Workshop on Research Issues in Data Engineering (RIDE 2001), Heidelberg, Germany, April 1-2, 2001, pp. 1-8, hereafter referred to as "XEM") and further in view of Swamy et al. (US Patent No. 6,874,141, filed Jun. 29, 2000 and issued Mar. 29, 2005, hereafter referred to as "Swamy") and Peter Buneman et al ("UnQL: A Query Language and Algebra for SemiStructured Data Based on Structural Recursion", The VLDB Journal, Issue No. 9, Springer-Verlag, © 2000, pp. 76-110, hereafter referred to as "Buneman").

**Regarding dependent claims 22-23 and 25-26,** SchemaMatching discloses matching source and target nodes and generating a transformation between nodes in phases 3 and 4 on page 2, discussing the matching likelihood of component pairs” (phase 3) and producing a “best matching plan” in phase 4. However, SchemaMatching does not explicitly disclose computing for a sequence of transformation operations, such as unfolding. Buneman, though, teaches the calculation of value equivalence in performing operations on trees in section “4.1 value Equivalence” starting on page 88, discussing the determination of a value equivalence for a data graph d and the resulting graph from an unfold operation in the first and second paragraphs under the section heading.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Buneman for the benefit of SchemaMatching in view of XEM, because to do so would have allowed one to implement a value-based semistructured data model, as taught by Buneman in last paragraph on page 77. These references were all applicable to the same field of endeavor, i.e., XML programming.

**Regarding dependent claim 24,** SchemaMatching does not explicitly disclose matching node labels. XEM, though, teaches the use of node labels in section “2.2 The DTD Data Model”, discussing a labeling for node attributes. It was implicit in node matching that at least one node attribute must be matched, and it was a obvious variant at the time of the invention as to which attribute (e.g. label name) was matched between nodes.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of XEM for the benefit of SchemaMatching in view of Buneman, because

Art Unit: 2162

to do so would have allowed a designer to change a DTD without requiring change of underlying XML data, as taught by XEM in top left paragraph of page 2. These references were all applicable to the same field of endeavor, i.e., XML programming.



***Conclusion***

9. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Art Unit: 2162

***Contact Information***

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert Stevens whose telephone number is (571) 272-4102. The examiner can normally be reached on M-F 6:00 - 2:30.


If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



Robert Stevens  
Examiner  
Art Unit 2162

October 4, 2006



SHAHID ALAM  
PRIMARY EXAMINER